Penglai: Verifiable and Scalable TEE system

Shanghai Jiao Tong University IPADS · Dong Du

FOSDEM 2021
Enclave/TEE: Trusted Execution Environment

• Enclave/TEE
  – A trusted execution environment (TEE) is a secure area of a main processor. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity —— Wikipedia

• Two major functionalities
  – Remote attestation: whether a remote node is the enclave with legal code
  – Isolation: untrusted SW/HW can not access enclave’s data

• Enclave’s capability: restrict data access
  – Data is only transferred among attested nodes
The Rising TEE

- Intel SGX/TDX
- AMD SEV
- ARM TrustZone
- Keystone, Penglai

- Cloud vendors utilize TEE/Enclave to protect data
  - 2018, Microsoft Azure proposes Confidential Computing based on Intel SGX
  - 2019, Amazon proposes Nitro Enclave to protect sensitive user data
  - 2020, Google Cloud introduces the Secure VM based on AMD SEV

- Confidential computing consortium
  - Arm, AMD, Intel, Redhat, Facebook, Google
  - Huawei, Ali Cloud, Tencent, Baidu, Byte dancing
Penglai: Verifiable and Scalable RISC-V Enclave

- Secure hardware extensions
  - sPMP (Supervisor-mode Physical Memory Protection)

- Security monitor
  - Lightweight software/firmware in RISC-V Machine-mode
  - Formal verification-oriented design
  - Remote attestation, runtime management and isolation

- Secure runtime frameworks
  - ARM PSA, global platform
  - Easy to port existing secure applications

RISC-V modes

- U-Mode
- S-Mode
- M-Mode
- Hardware
Penglai Enclave: HW-SW Co-design for Security

- RISC-V Core
- Enclave
- Monitor
- PMP/sPMP
- RISC-V Core
- PMP/sPMP
- RISC-V Core
- Memory (DRAM, SRAM)
- sPMP is an extension proposed by Penglai

- PMP/sPMP
- PMP/sPMP
- PMP/sPMP
- IOPMP
- IOPMP
- IOPMP
- IOPMP
- Flash etc.
- External storage
- True RNG
- OTP
- Peripherals

Each IOPMP node can be connected with multiple IO devices.

IOPMP & PMP are configured by monitor.

sPMP is an extension proposed by Penglai.
Software Architecture

• Based on RISC-V-v1.10 spec

• Components: Monitor and Enclave App
  – Monitor: sPMP/PMP/IOPMP configurations, isolation, enclave management
  – Enclave App is responsible for executing tasks

• Enclave App includes Host Enclave App and Secure Enclave App
  – Host Enclave App: security non-sensitive tasks in REE (rich execution environment)
  – Secure Enclave App: security sensitive tasks in TEE
    • Service Enclave App: secure storage, encryption, etc.

• Designed for both MMU and non-MMU (e.g., MCU) devices

• Formal verification-oriented design
Penglai Architecture on MMU Chips

- **Host Enclave App**
  - Non-sensitive code/data

- **Secure Enclave App**
  - Sensitive code/data

- **Enclave Service**
  - FS code/data

- **OS**
  - Kernel module
    - SBI interface wrapper

- **Secure communication channel**

- **User**
  - Host Enclave App
  - Secure Enclave App
  - Enclave Service

- **Supervisor**
  - Monitor call
  - ioctl()
  - Monitor call

- **Machine**
  - Security Monitor
    - Enclave management

- **Hypervisor (Unused)**

- **Hardware property**
  - Memory isolation (sPMP/PMP)
  - Cache partition
  - Trusted
    - Untrusted
Penglai Architecture on non-MMU Chips (M+U)

User mode

Normal RTOS Regions/
Bare-metal regions

Host SDK

Task

Secure Service Regions

Remote Attest
Secure Storage
TLS

Some FreeRTOS instructions are replaced to ecall

Secure Monitor

Emulate (privileged) instructions in monitor to avoid modifications on FreeRTOS

Penglai Secure Bootloader

Region Mgmt. / Comm.

PMP mgmt.

Trust App

Trust App1

Trust SDK
Penglai Architecture on non-MMU Chips (M+S+U)

- User mode
  - Normal RTOS Regions/
    Bare-metal regions
  - Task
    - Host SDK

- Supervisor mode
  - Modification on RTOS is minor
  - RTOS task mgr. /ISR
    - Host Driver

- Machine mode
  - Secure Monitor
    - PMP mgmt.
    - Region Mgmt. / Comm.
    - Penglai Secure Bootloader

- Secure Service Regions
  - Remote Attest
  - Secure Storage
  - TLS

- Trust App
  - Trust App1
  - Trust SDK
  - Driver to handle the requests from SDK

Modification on RTOS is minor.
Formal Verification

• Motivation
  – Hardware provides basic primitives: isolation (PMP, sPMP), cache partition, and others
  – Software monitor is responsible for implementing others
    • The only software TCB, security sensitive
    • Formal methods!
Formal Verification

- **Pangolin framework: formal verification**
  - Formal specification describing functionalities
  - Verify functionalities and higher security properties
  - Based on model checking and symbolic execution
Formal Verification-Oriented Design

- Big monitor lock
  - Sufficient for monitor yet more verifiable [1]

- Eliminate/restrict unbounded loops

- Verification friendly interface
  - Constrained pointers in arguments

Formal Verification

• **Verified modules**
  – RISC-V boot process, IPC calls, helper functions

• **Future work**
  – Enclave management
  – Enclave fork
  – Others
Secure Functionalities

- Memory isolation
- Interrupt isolation
- Secure storage
- Secure usage of peripherals
Memory Isolation

- Utilize sPMP + PMP for enclave memory isolation

![Diagram showing memory isolation between different enclaves in various modes of operation.](image-url)
sPMP (S-mode PMP)

- For IoT devices (MMU-less)
  - Enable S-mode OS to limit the physical addresses accessible by U-mode software

```
  U  Apps
     --
  S  RTOS
     --
  M  Monitor
```

- No isolation
- PMP isolation

IoT (MMU-less)
sPMP (S-mode PMP)

- S-mode virtualization for scalable enclaves

(a) PMP-based isolation

(b) sPMP
sPMP (S-mode PMP)

- sPMP entries
  - 8-bit configuration register (SMAP enabled by default)
  - XLEN-bit address register

Refer the proposal in RISC-V/TEE group for details.
Interrupt Isolation

- **Goal:** interrupts are only visible to the target Enclave Apps

- **Controllers provide different granularities on configuration**
  - PLIC: configure whether external interrupts should be directed to S-mode
  - CLIC/ECLIC: could configure whether individual interrupt should be directed to S-mode

- **Different isolation mechanisms for different controllers**
Interrupt Isolation: Platform-Level Interrupt Controller

- External interrupts are always trapped into M-mode
- Monitor is responsible for interrupt redirection

- External interrupt, always trap into machine mode
- S-mode/u-mode
- Machine mode
- Secure Monitor

Interruptions are assigned to enclaves, which are configured during initialization through a configuration file.

The entry of interrupt handler is specified through monitor calls (or it could allow monitor directly jumps to the entry in stvec).

Ack interrupt, query which enclave is connected to the interrupt, and jump to the handler.
Interrupt Isolation: Core-Local Interrupt Controller

- If s-mode exists and the interrupt is related to the running enclave, it is handled by the enclave directly.

**Case I**
- Enclave A
  - Running
  - Inturrupts (belong Enclave A)
- Enclave B
  - S-Mode Handler

**Case II**
- Enclave A
  - Running
  - Inturrupts (belong Enclave B)
- Enclave B
  - S-Mode Handler

If the interrupts are not assigned to the running enclave, it will trap into the monitor, which will redirect interrupts to the target enclave.

Relations between interrupts and enclaves are configured by monitor during initialization through a configuration file. When switching to a new enclave, all the interrupts assigned to enclave are configured to be directed to S-mode, and others will be trapped into the monitor.
Secure Storage

Secure storage is provisioned by specific Service Enclave

Enclave App invokes the Storage Service Enclave through IPC. According to different scenarios, callers can use Global Platform API or PSA API.

Secure storage guarantees privacy and integrity protection on data, and can defend replay attacks.

For MCU devices, storage is usually the fixed region in internal flash, which is protected by PMP.

For non-MCU devices, our solutions utilize RPMB regions in EMMC/UFS as the storage (under development now)
Secure Usage of Peripherals

Restricting the requests issued by RISC-V Core through PMP/sPMP

Restricting the (DMA) requests issued by device through IOPMP (WIP)

- Each I/O device should be assigned with a master id
- Configure the permissions of the master id through IOPMP
Scenarios: Secure Communication

• TEE NSSDK/TEE_TA SDK: functionalities to allow communication between Enclaves and Untrusted Apps, and among Enclaves
• Support mainstream crypto algorithms for encryption/description/hashing/integrity.
• TEE-enhanced SSL/TLS protocols
• Support both PSA and GP API
Penglai: Verifiable and Scalable TEE

• Verifiable
  – Formal verification-oriented design
  – Pangolin framework

• Scalable
  – Utilize scalable hardware isolation mechanism: PMP + S-mode PMP
  – Running up to 1000 (concurrently) enclaves in Qemu/FPGA

• Security functionalities
  – Memory isolation, secure storage, interrupts, and peripherals

• Open-sourced
  – https://github.com/Penglai-Enclave/Penglai-Enclave
Thanks